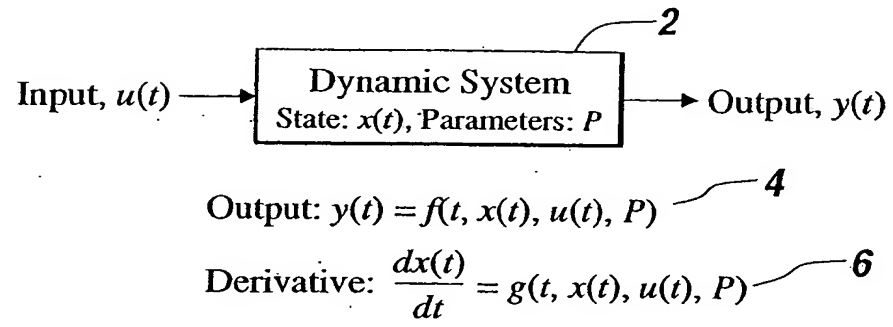

Fig. 1A

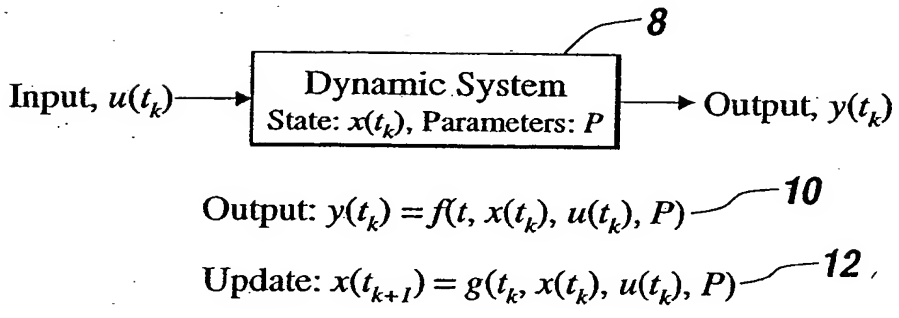

Fig. 1B

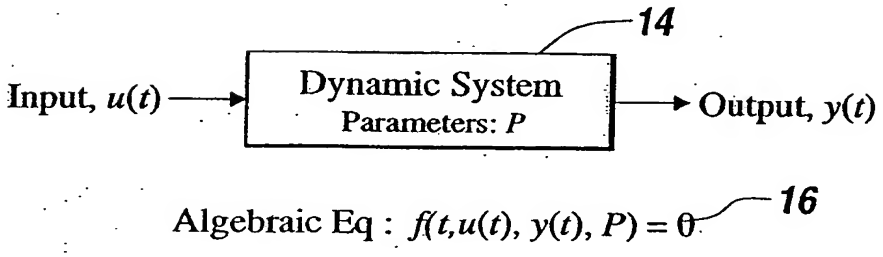

Fig. 1C

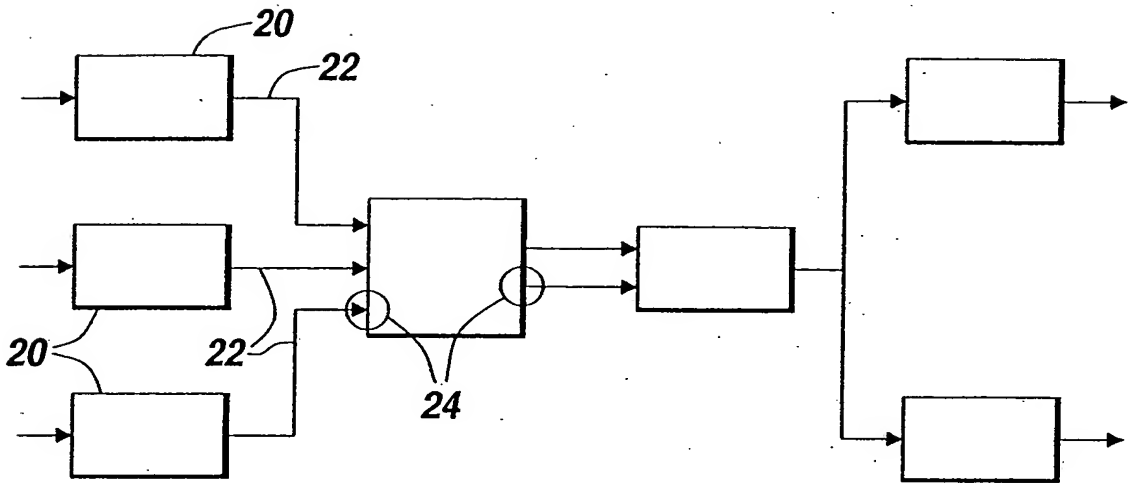

Fig. 2

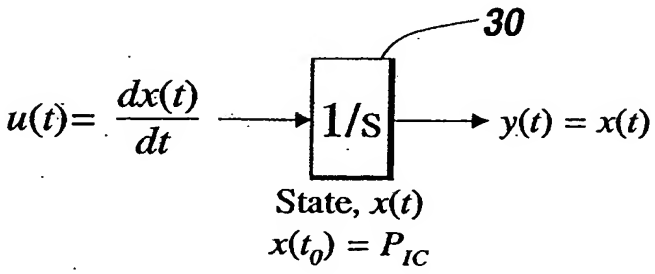

Fig. 3

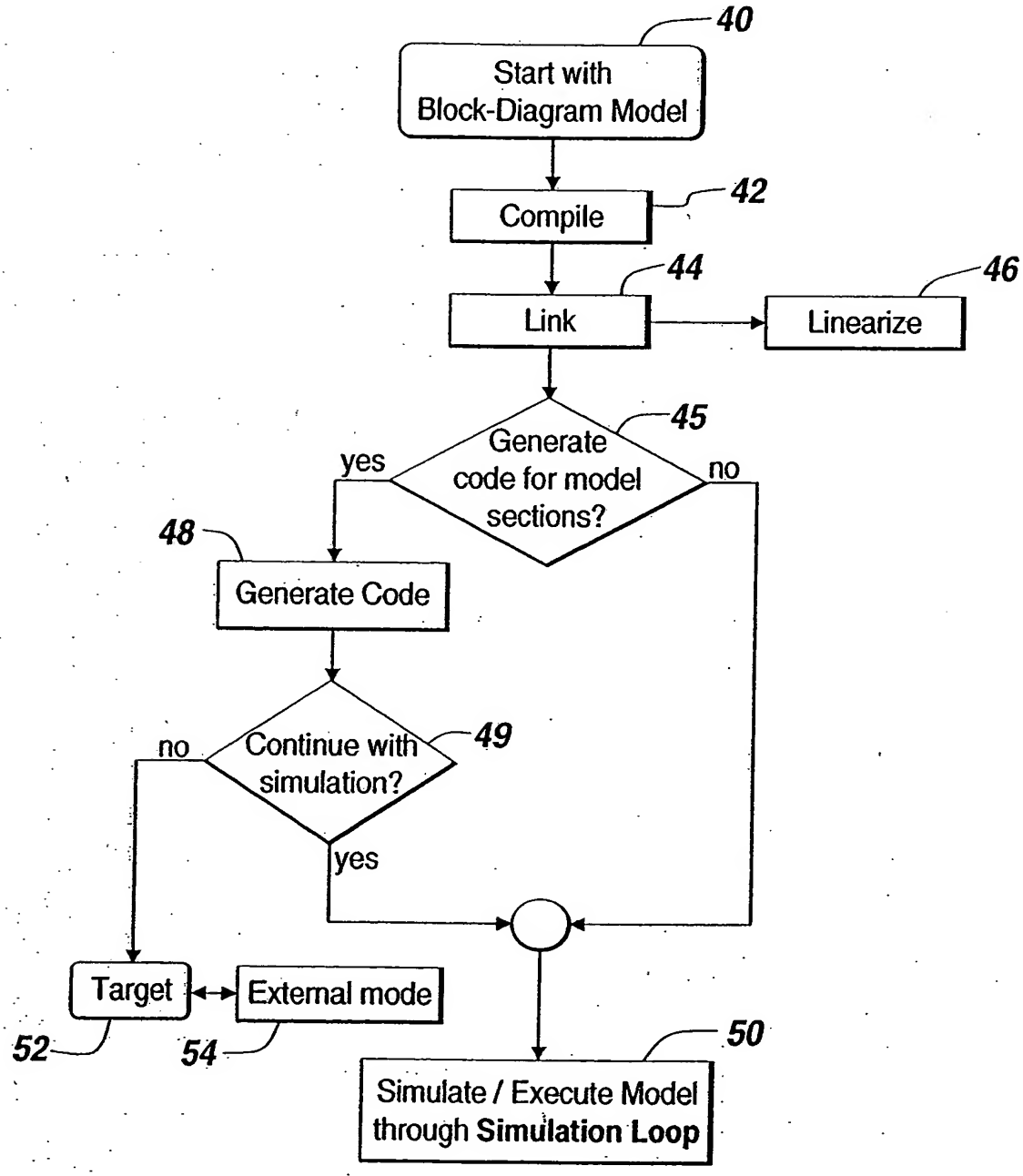
40
Start with Block-Diagram Model
Compile
42
Link
44
Linearize
46
Generate code for model sections?
45
yes
no
48
Generate Code
Continue with simulation?
49
no
yes
Target
External mode
52
54
Simulate / Execute Model through Simulation Loop
50

Fig. 4

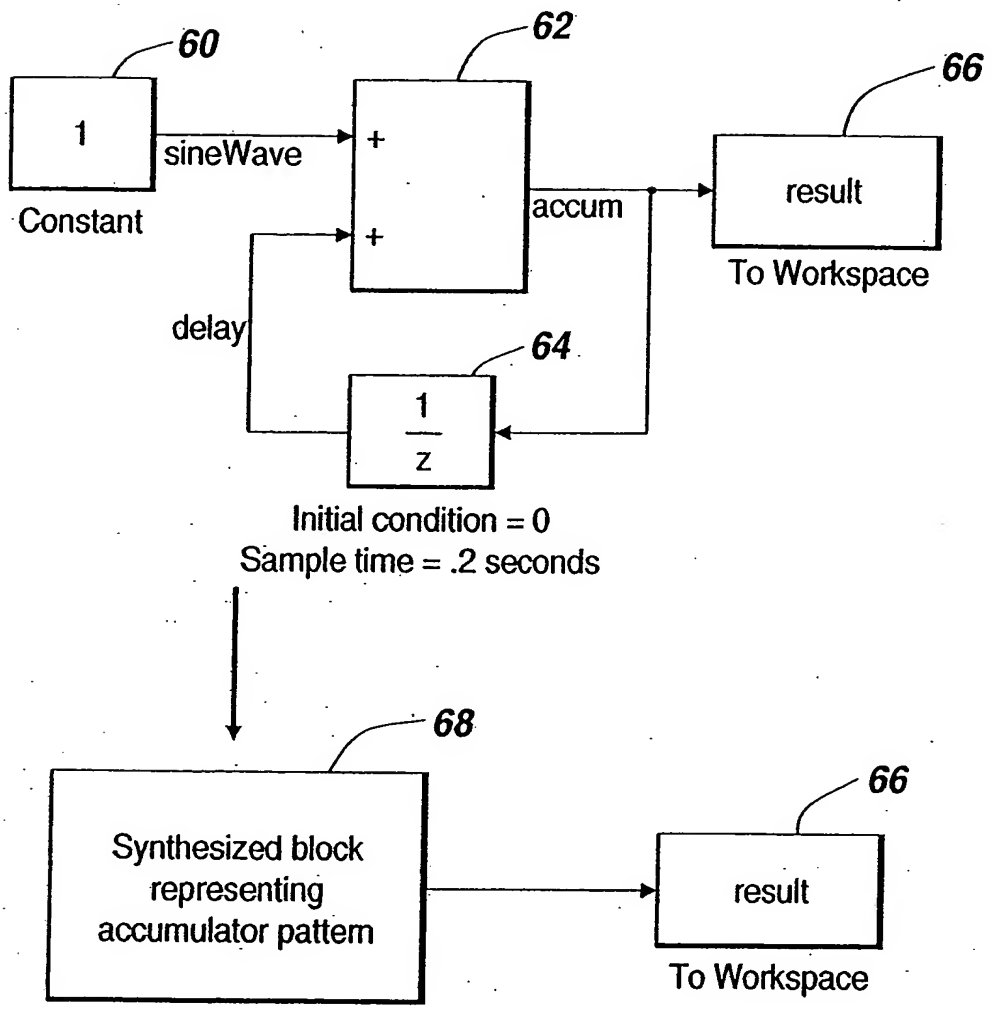
60
62
66
1
sineWave
+
+
accum
result
Constant
To Workspace
delay
64
1
z
Initial condition = 0
Sample time = .2 seconds
68
66
Synthesized block representing accumulator pattern
result
To Workspace

Fig. 5

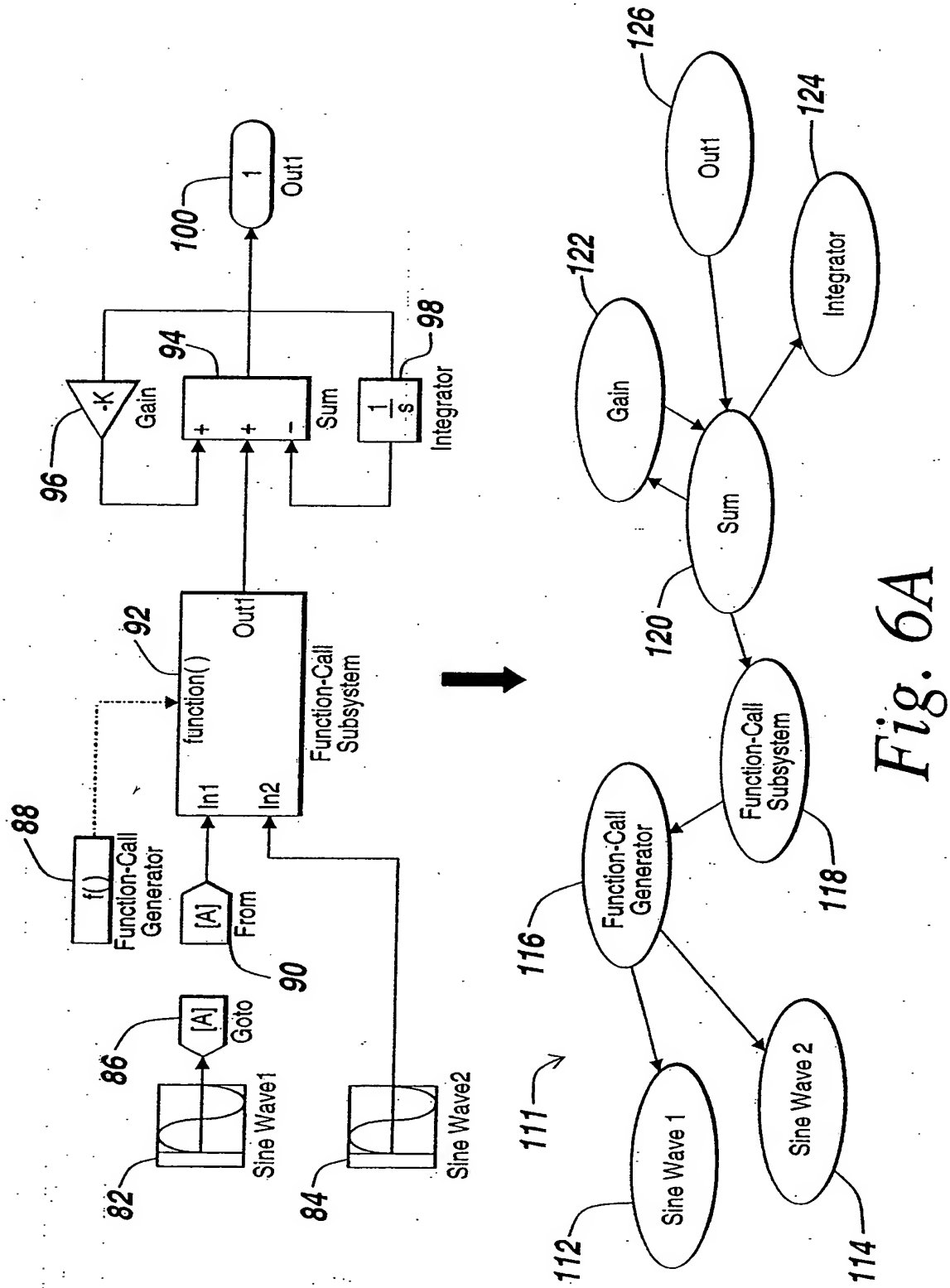
Out1
1
100
Integrator
1/s
98
Sum
+
+
-
94
Gain
-K
96
Out1
function()
In1
In2
92
Function-Call Subsystem
f()
Function-Call Generator
88
[A]
From
90
[A]
Goto
86
Sine Wave1
82
Sine Wave2
84
Out1
126
Integrator
124
Gain
122
Sum
120
Function-Call Subsystem
118
Function-Call Generator
116
111
Sine Wave 1
112
Sine Wave 2
114

Fig. 6A

Sorted List:

0:0 Sine Wave 1
0:1 Sine Wave 2
0:2 Function-Call Generator
0:3 Function-Call Subsystem
0:4 Integrator
0:5 Gain (algebraic id 0#1)
0:6 Sum (algebraic variable for id 0#1)
0:7 Out1

Fig. 6B

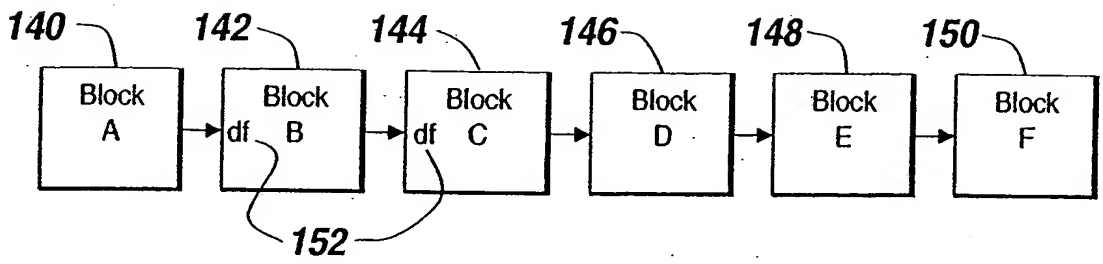

Fig. 7A

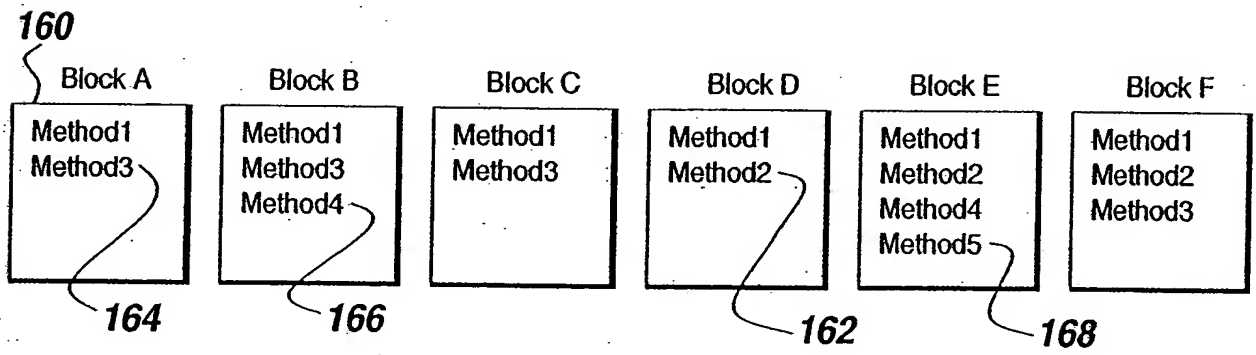

Fig. 7B

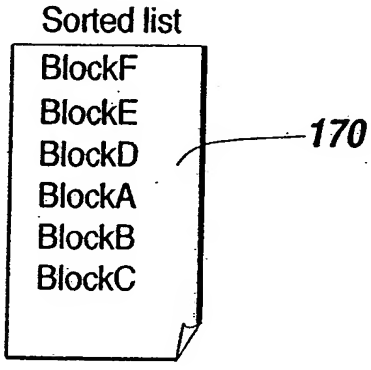

Fig. 7C

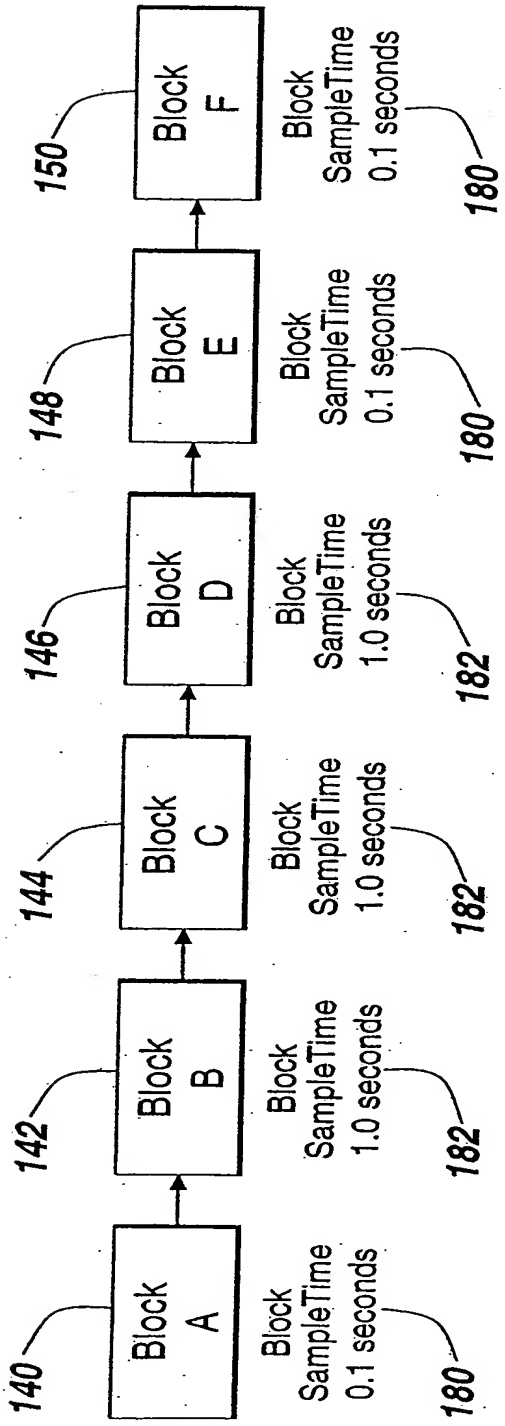
Block A
Block SampleTime 0.1 seconds
140
180
Block B
Block SampleTime 1.0 seconds
142
182
Block C
Block SampleTime 1.0 seconds
144
182
Block D
Block SampleTime 1.0 seconds
146
182
Block E
Block SampleTime 0.1 seconds
148
180
Block F
Block SampleTime 0.1 seconds
150
180

Fig. 8

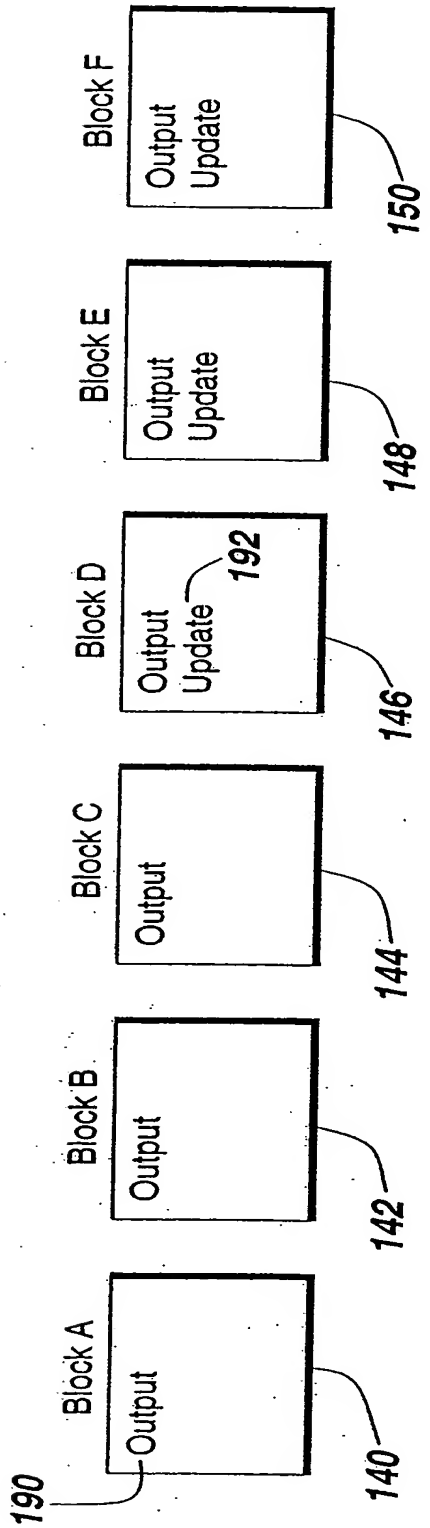
Block A
Output
190
140
Block B
Output
142
Block C
Output
144
Block D
Output Update
192
146
Block E
Output Update
148
Block F
Output Update
150

Fig. 9

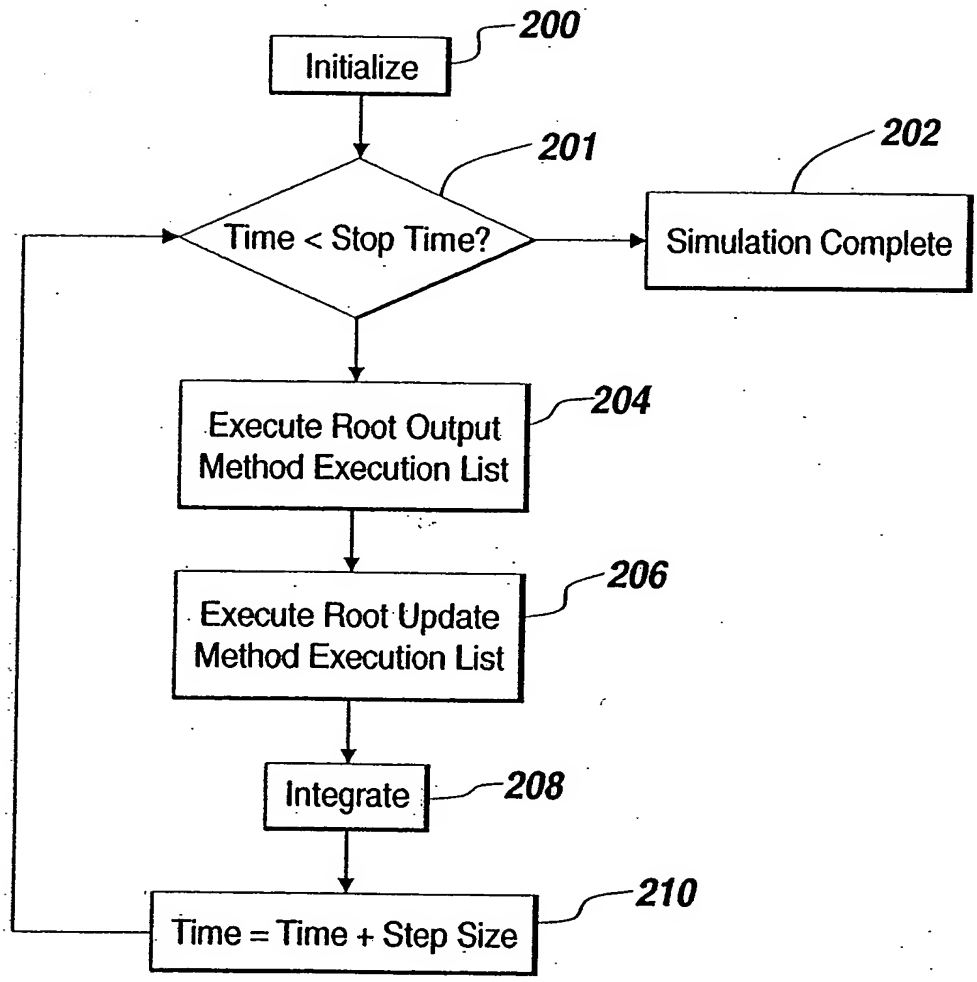
200
Initialize
201
Time < Stop Time?
202
Simulation Complete
204
Execute Root Output Method Execution List
206
Execute Root Update Method Execution List
208
Integrate
210
Time = Time + Step Size

Fig. 10

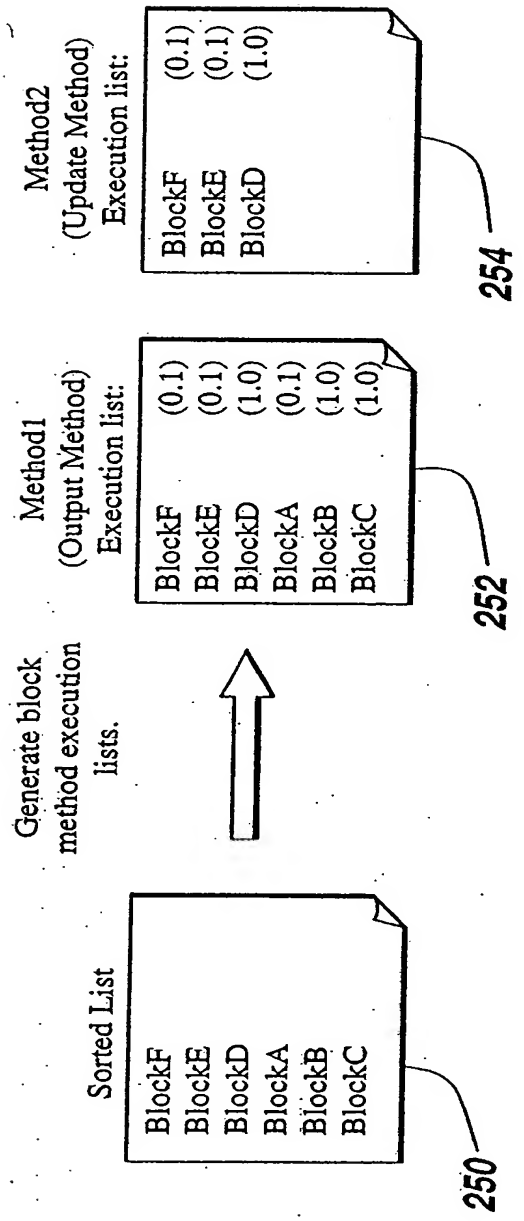
Sorted List
BlockF
BlockE
BlockD
BlockA
BlockB
BlockC
250
Generate block method execution lists.
Method1
(Output Method)
Execution list:
BlockF (0.1)
BlockE (0.1)
BlockD (1.0)
BlockA (0.1)
BlockB (1.0)
BlockC (1.0)
252
Method2
(Update Method)
Execution list:
BlockF (0.1)
BlockE (0.1)
BlockD (1.0)
254

Fig. 11A

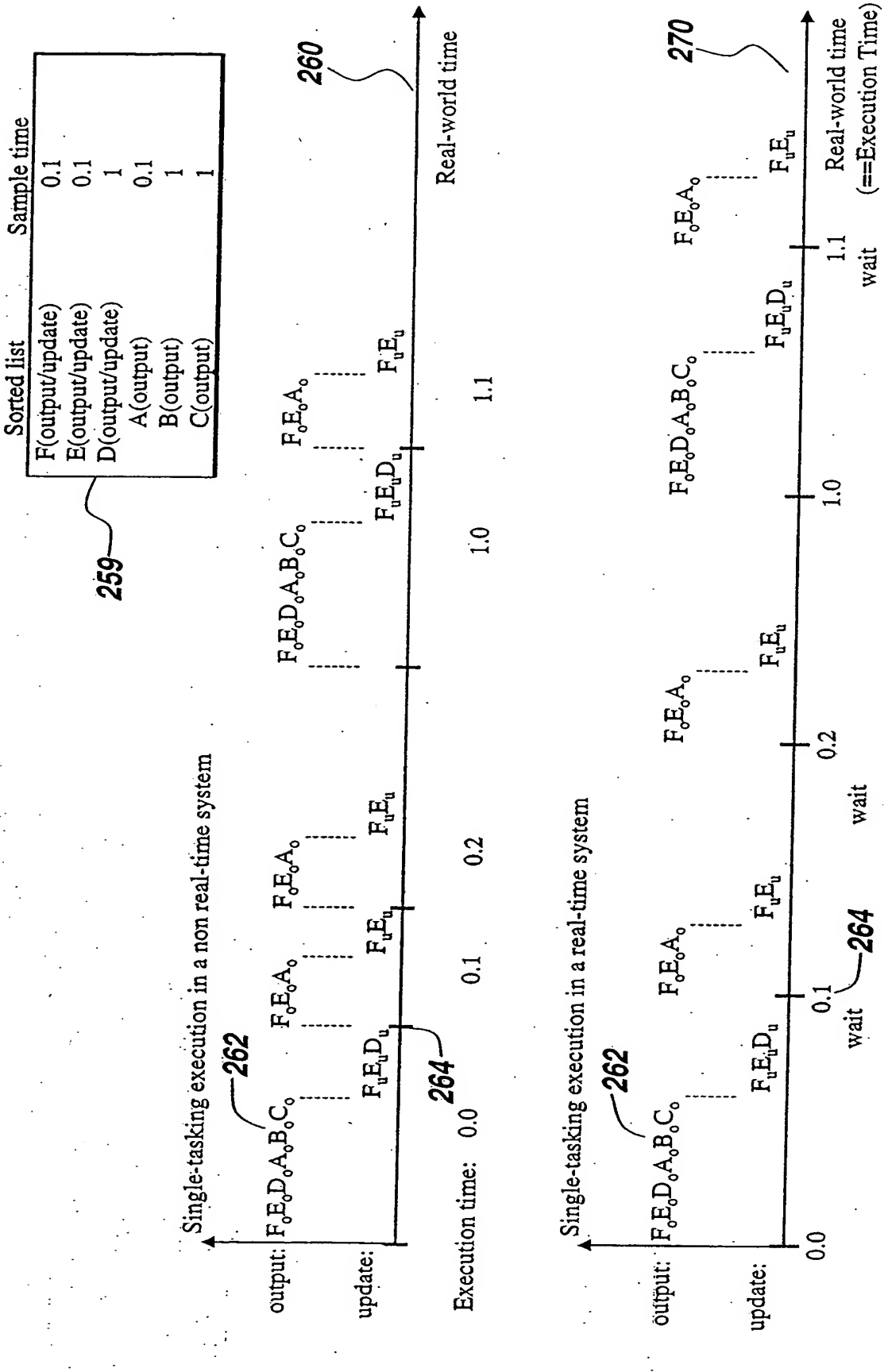
Sorted list
Sample time
F(output/update) 0.1
E(output/update) 0.1
D(output/update) 1
A(output) 0.1
B(output) 1
C(output) 1
259
260
Real-world time
Single-tasking execution in a non real-time system
262
output:
update:
Execution time:
0.0
0.1
0.2
1.0
1.1
264
Single-tasking execution in a real-time system
270
wait
Real-world time (==Execution Time)

Fig. 11B

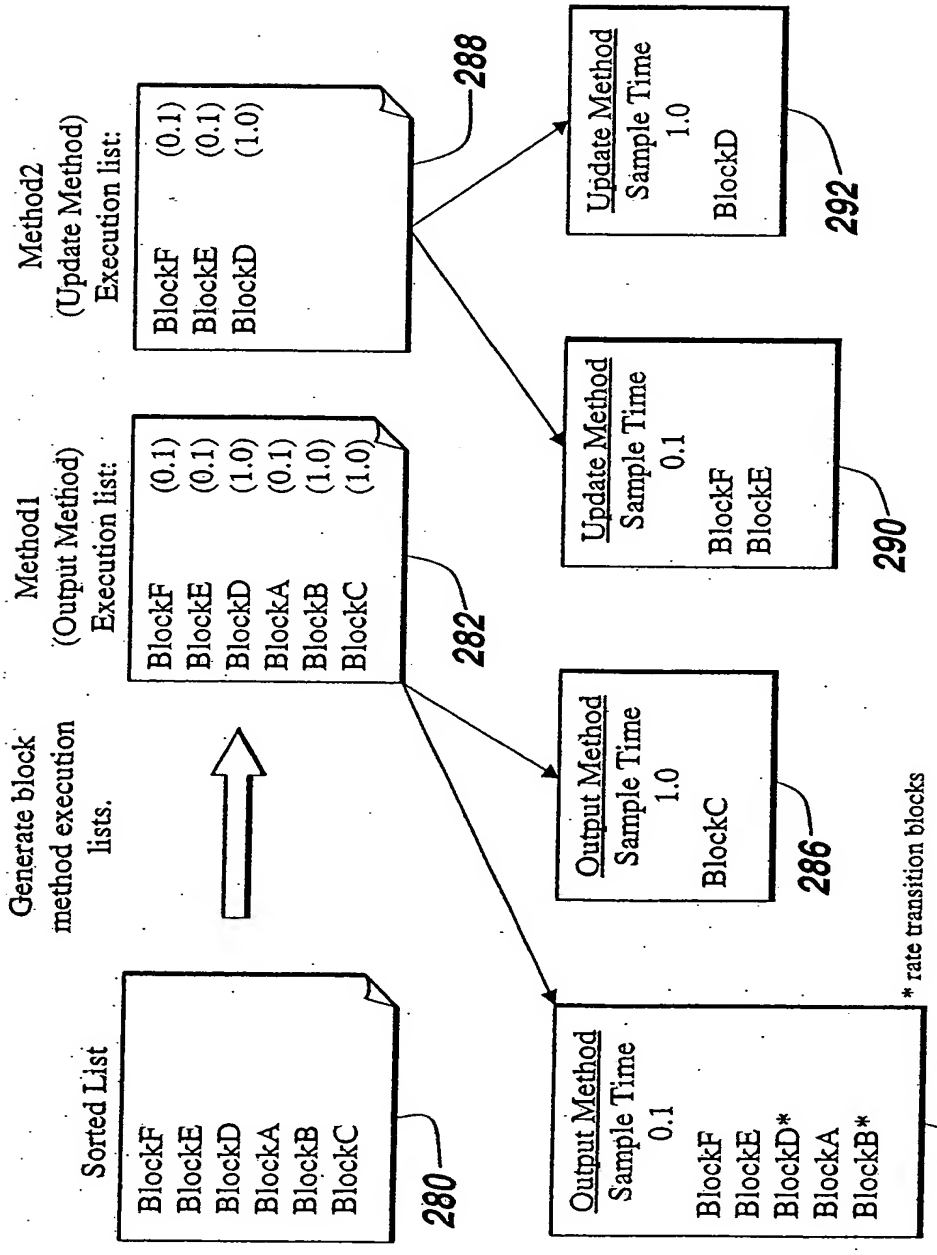
Sorted List
BlockF
BlockE
BlockD
BlockA
BlockB
BlockC
280
Generate block method execution lists.
Method1
(Output Method)
Execution list:
BlockF (0.1)
BlockE (0.1)
BlockD (1.0)
BlockA (0.1)
BlockB (1.0)
BlockC (1.0)
282
Method2
(Update Method)
Execution list:
BlockF (0.1)
BlockE (0.1)
BlockD (1.0)
288
Output Method
Sample Time
0.1
BlockF
BlockE
BlockD*
BlockA
BlockB*
284
* rate transition blocks
Output Method
Sample Time
1.0
BlockC
286
Update Method
Sample Time
0.1
BlockF
BlockE
290
Update Method
Sample Time
1.0
BlockD
292

Fig. 12A

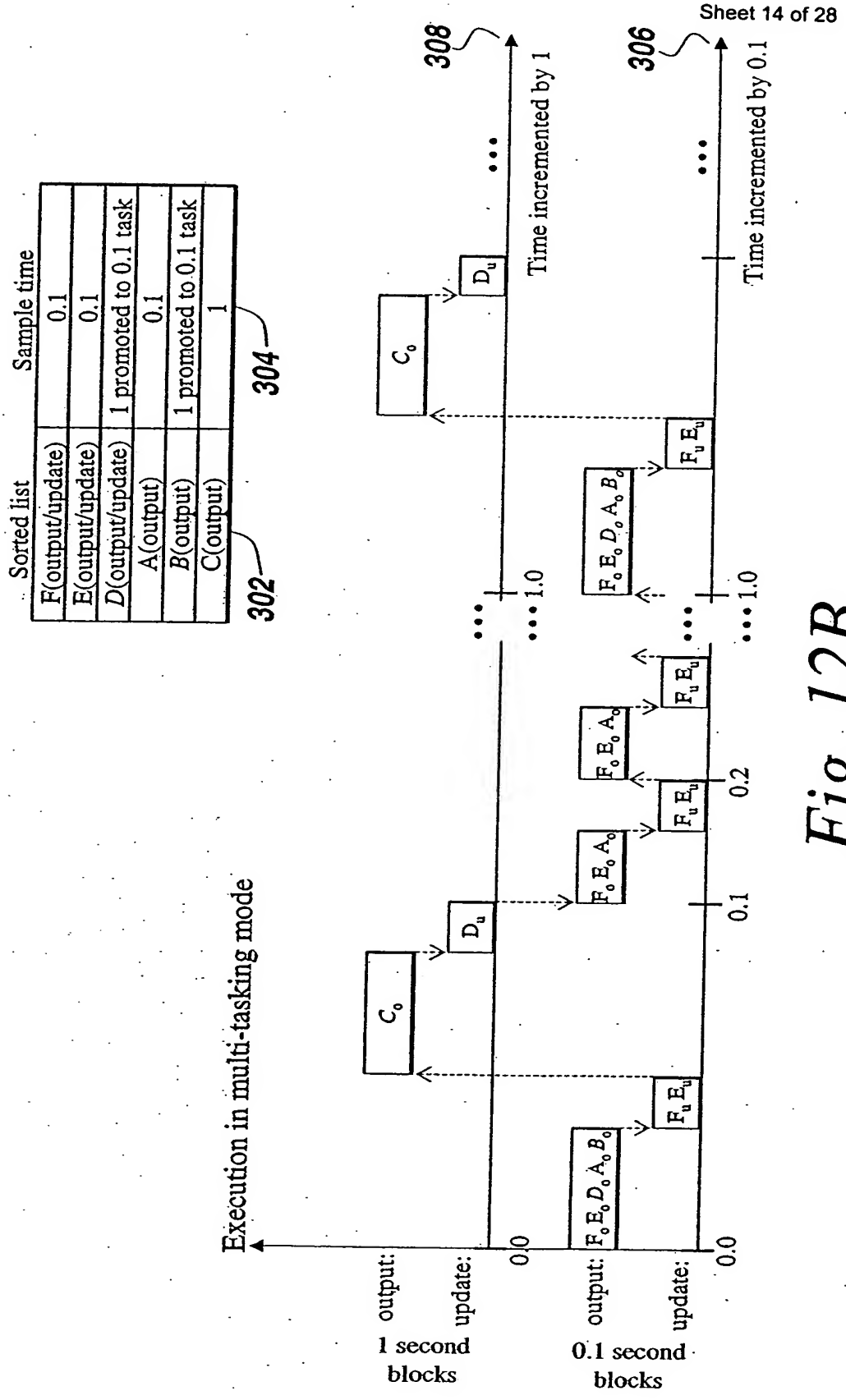
Sorted list
Sample time
F(output/update) 0.1
E(output/update) 0.1
D(output/update) 1 promoted to 0.1 task
A(output) 0.1
B(output) 1 promoted to 0.1 task
C(output) 1
302
304
308
306
Time incremented by 1
Time incremented by 0.1
Execution in multi-tasking mode
output:
update:
1 second blocks
0.1 second blocks
0.0
0.1
0.2
1.0

Fig. 12B

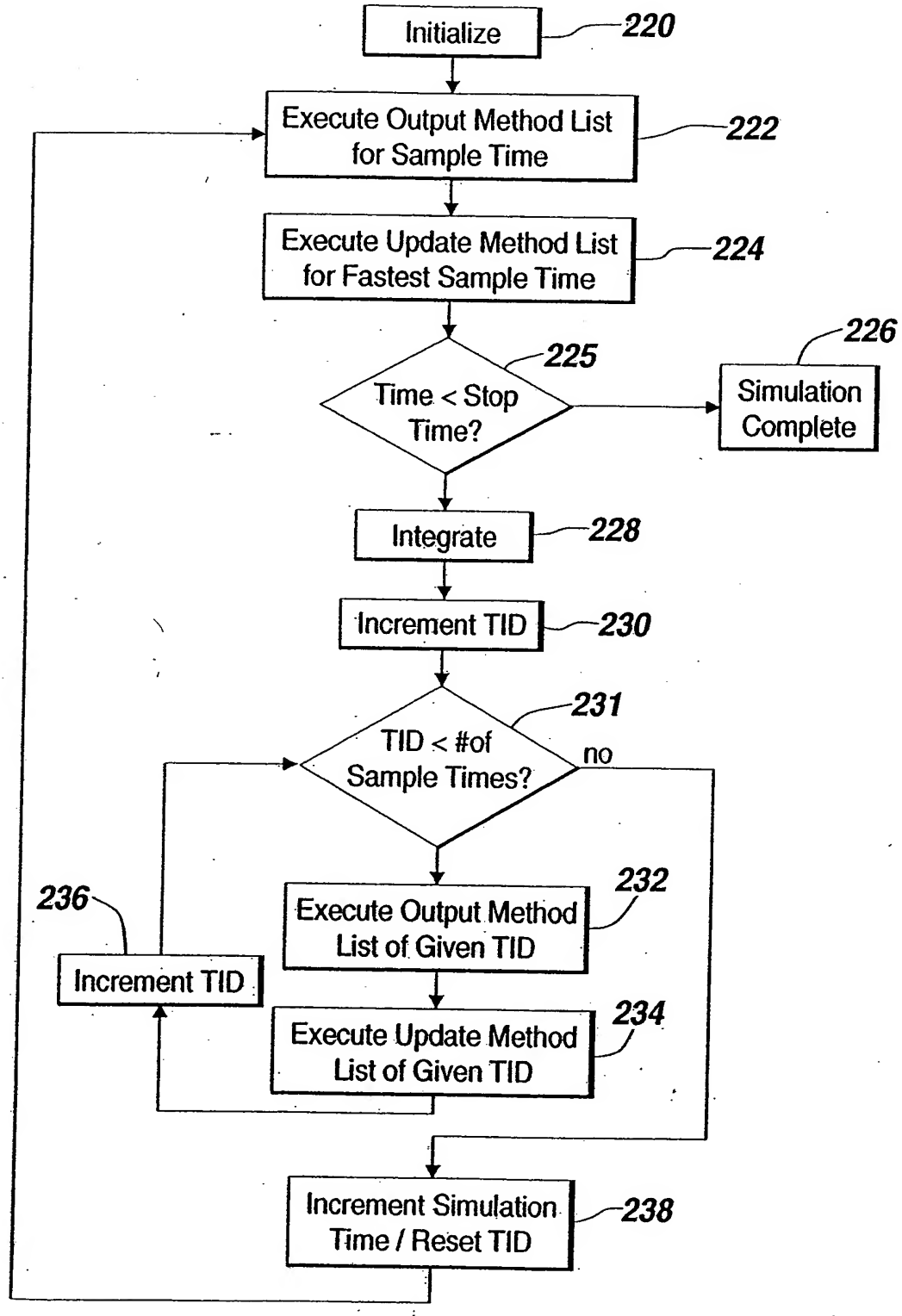
Initialize
220
Execute Output Method List for Sample Time
222
Execute Update Method List for Fastest Sample Time
224
Time < Stop Time?
225
Simulation Complete
226
Integrate
228
Increment TID
230
TID < #of Sample Times?
231
no
Execute Output Method List of Given TID
232
Execute Update Method List of Given TID
234
Increment TID
236
Increment Simulation Time / Reset TID
238

Fig. 13

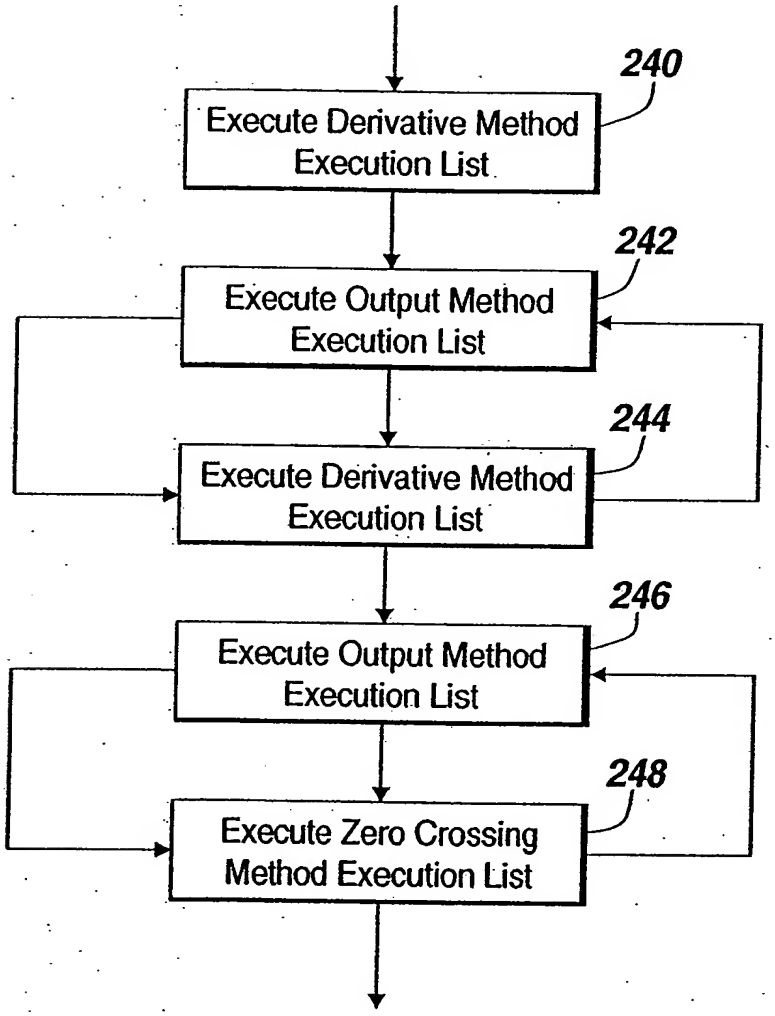
240
Execute Derivative Method Execution List
242
Execute Output Method Execution List
244
Execute Derivative Method Execution List
246
Execute Output Method Execution List
248
Execute Zero Crossing Method Execution List

Fig. 14

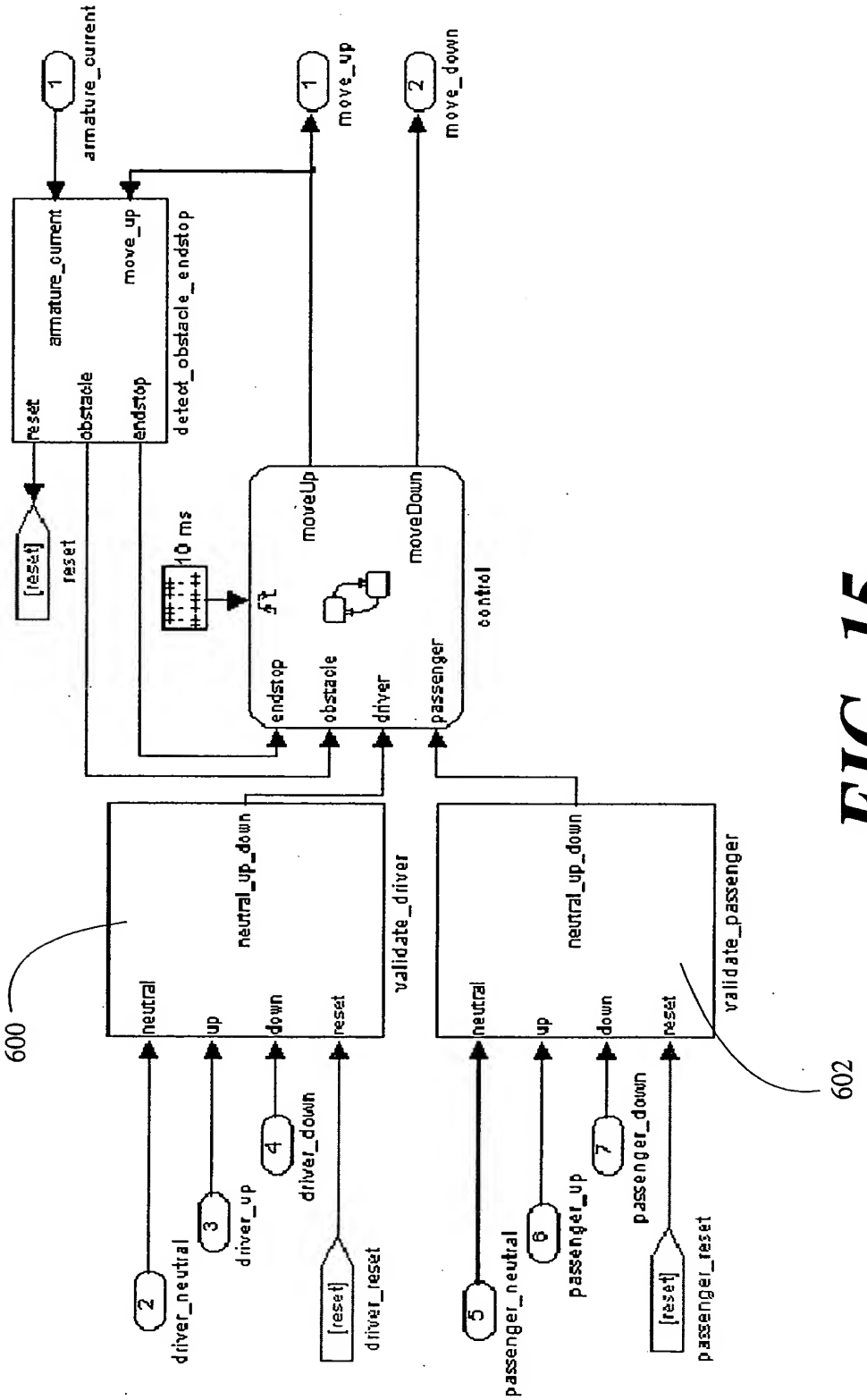
armature_current
move_up
move_down
armature_current
move_up
reset
obstacle
endstop
detect_obstacle_endstop
[reset]
reset
10 ms
moveUp
moveDown
endstop
obstacle
driver
passenger
control
600
neutral_up_down
validate_driver
neutral
up
down
reset
driver_neutral
driver_up
driver_down
driver_reset
602
neutral_up_down
validate_passenger
passenger_neutral
passenger_up
passenger_down
passenger_reset

FIG. 15

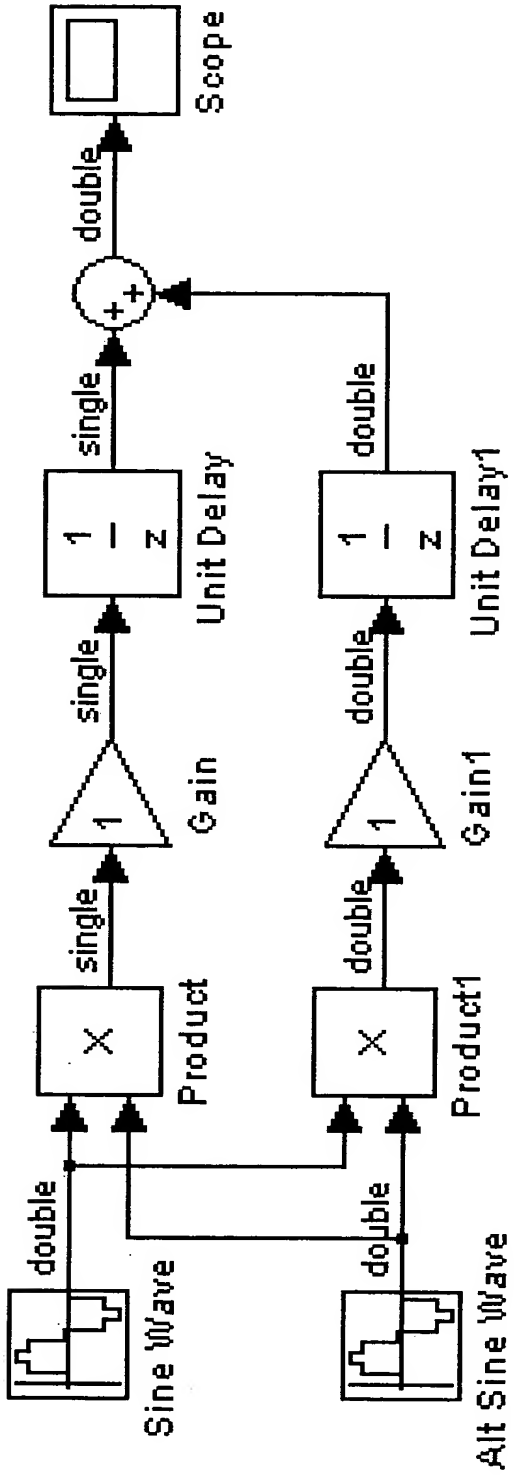
Scope
double
+
+
single
1
z
Unit Delay
double
1
z
Unit Delay1
single
1
Gain
double
1
Gain1
single
X
Product
double
X
Product1
double
double
Sine Wave
Alt Sine Wave

FIG. 16

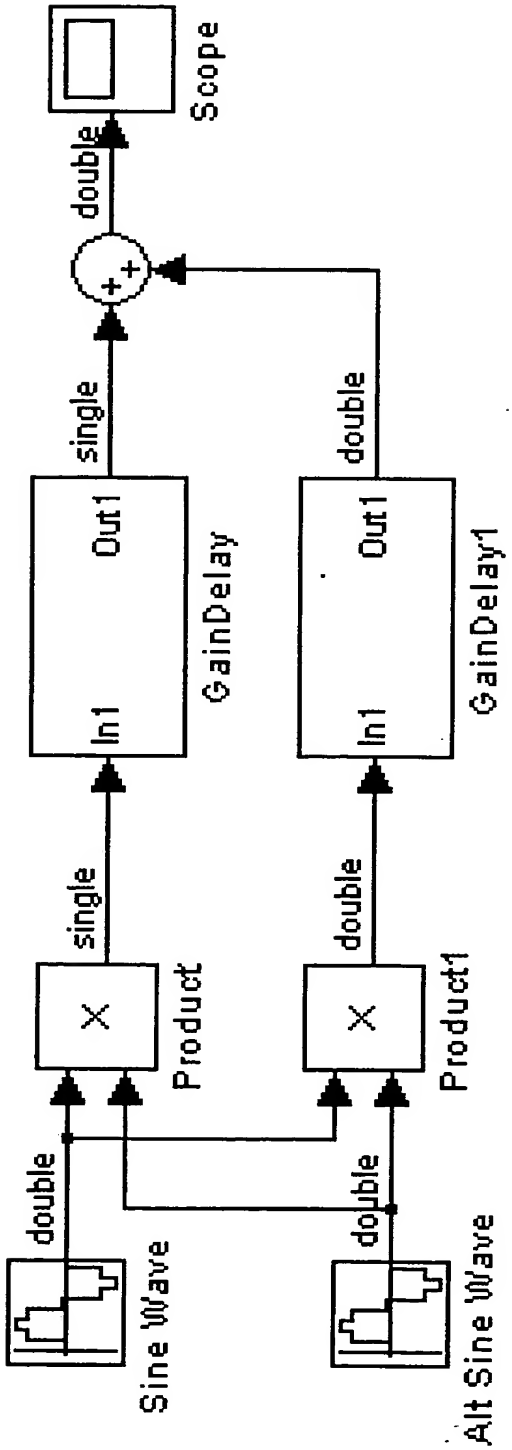
Scope
double
single
Out1
In1
GainDelay
double
Out1
In1
GainDelay1
single
double
X
Product
X
Product1
double
double
Sine Wave
Alt Sine Wave

FIG. 17

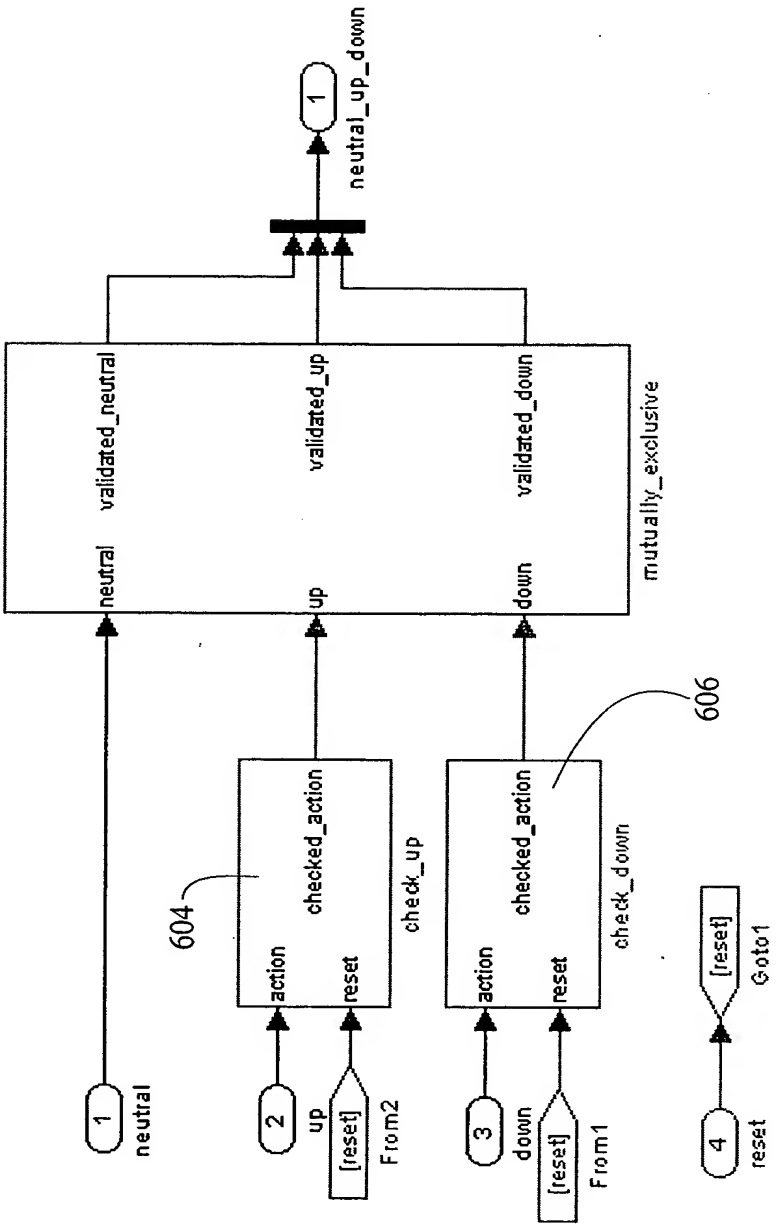
1
neutral_up_down
validated_neutral
validated_up
validated_down
mutually_exclusive
neutral
up
down
604
606
checked_action
action
reset
check_up
check_down
1
neutral
2
up
[reset]
From2
3
down
[reset]
From1
4
reset
[reset]
Goto1

FIG. 18

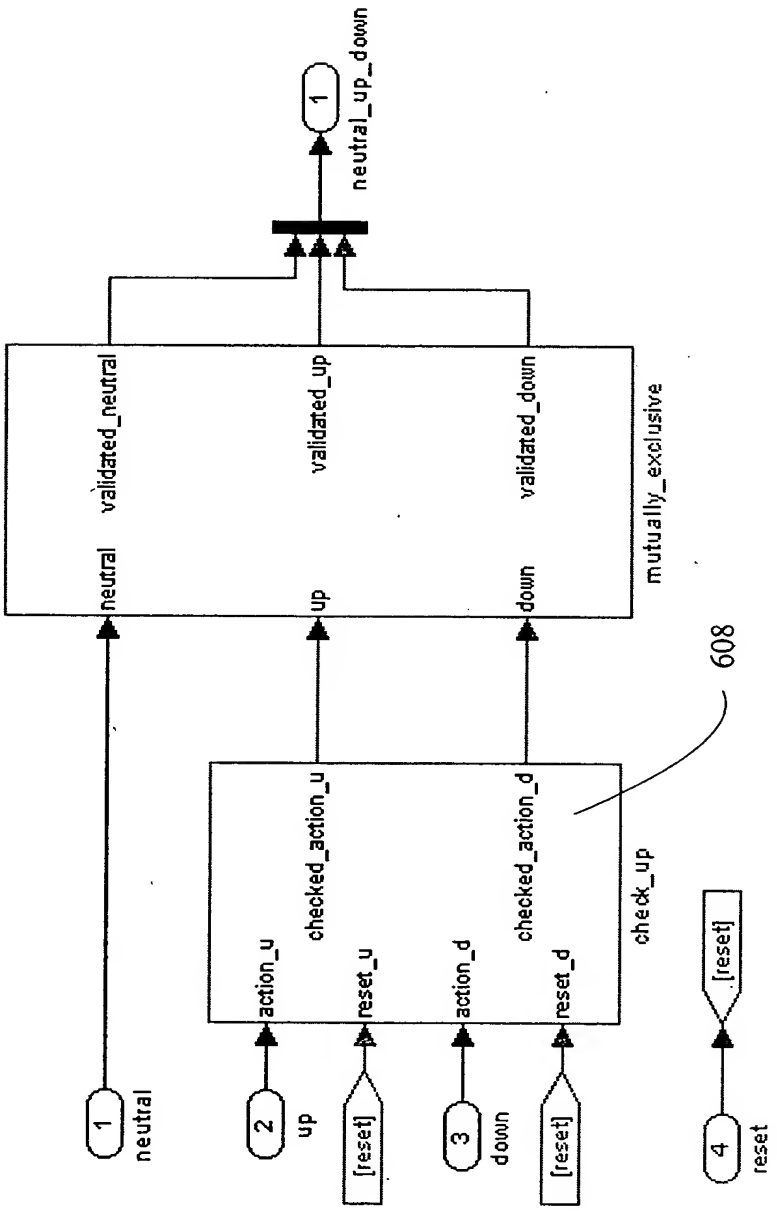
1
neutral_up_down
validated_neutral
validated_up
validated_down
mutually_exclusive
neutral
up
down
608
checked_action_u
checked_action_d
check_up
action_u
reset_u
action_d
reset_d
1
neutral
2
up
[reset]
3
down
[reset]
[reset]
4
reset

FIG. 19

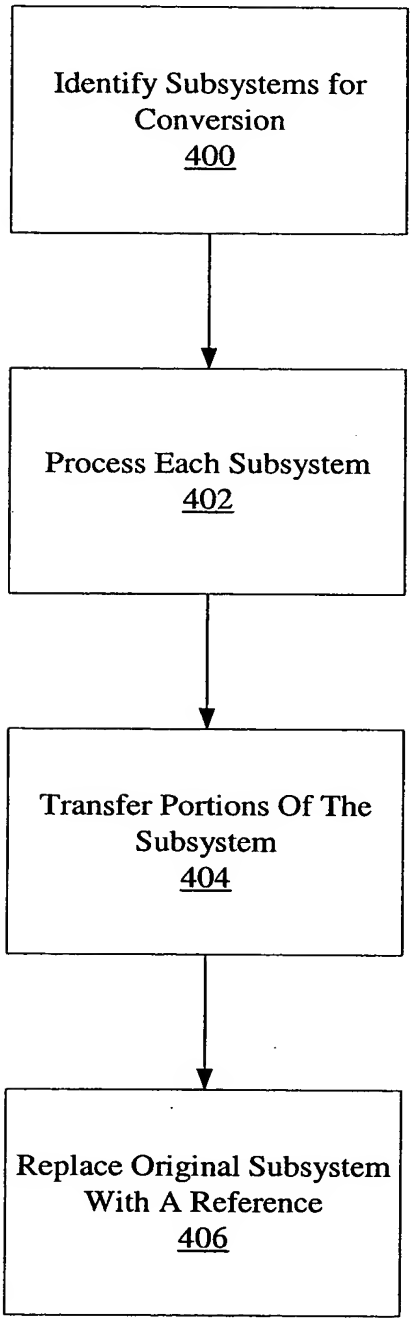
Identify Subsystems for Conversion
400
Process Each Subsystem
402
Transfer Portions Of The Subsystem
404
Replace Original Subsystem With A Reference
406

FIG 20

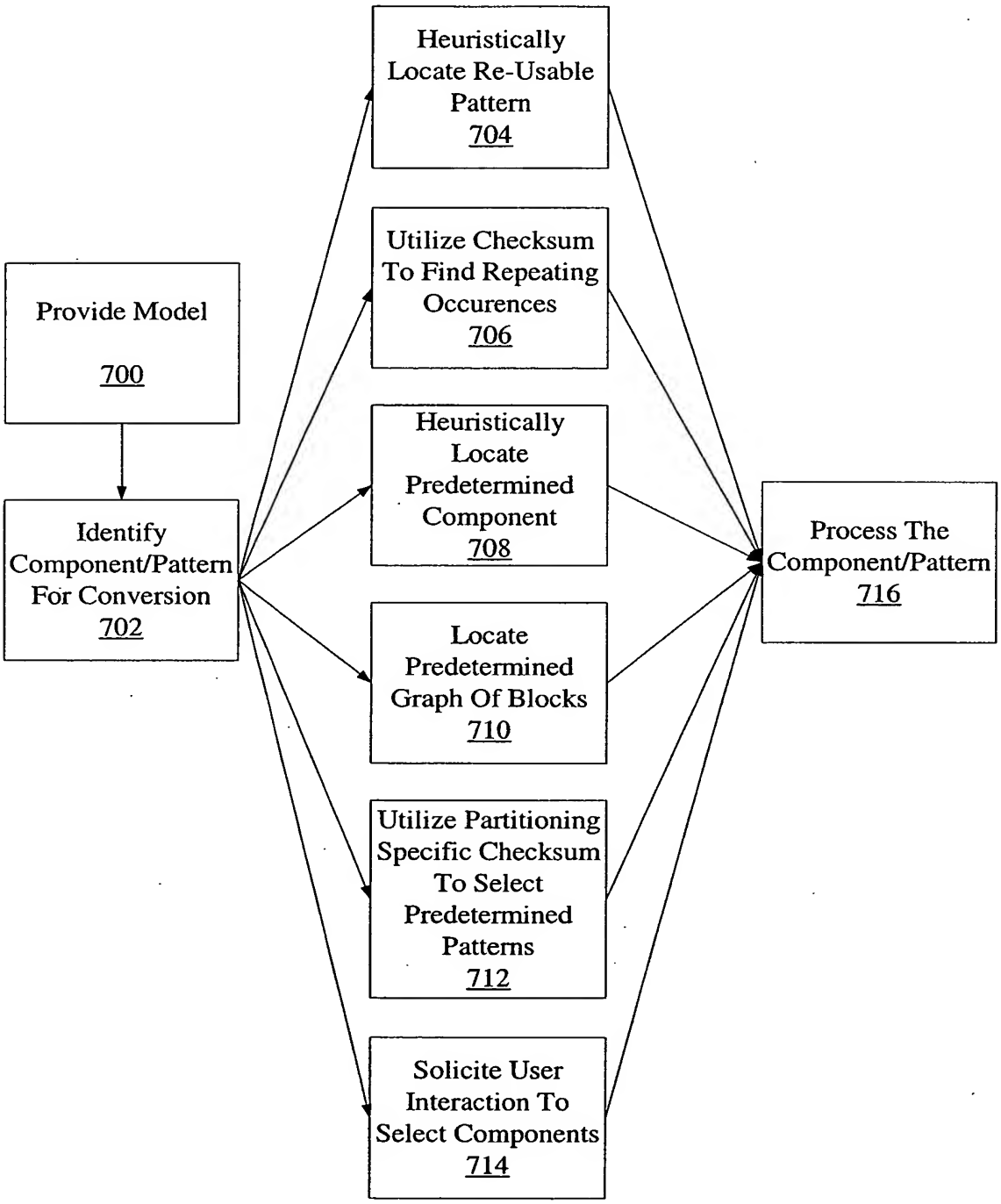
Provide Model
700
Identify Component/Pattern For Conversion
702
Heuristically Locate Re-Usable Pattern
704
Utilize Checksum To Find Repeating Occurences
706
Heuristically Locate Predetermined Component
708
Locate Predetermined Graph Of Blocks
710
Utilize Partitioning Specific Checksum To Select Predetermined Patterns
712
Solicite User Interaction To Select Components
714
Process The Component/Pattern
716

FIG 21

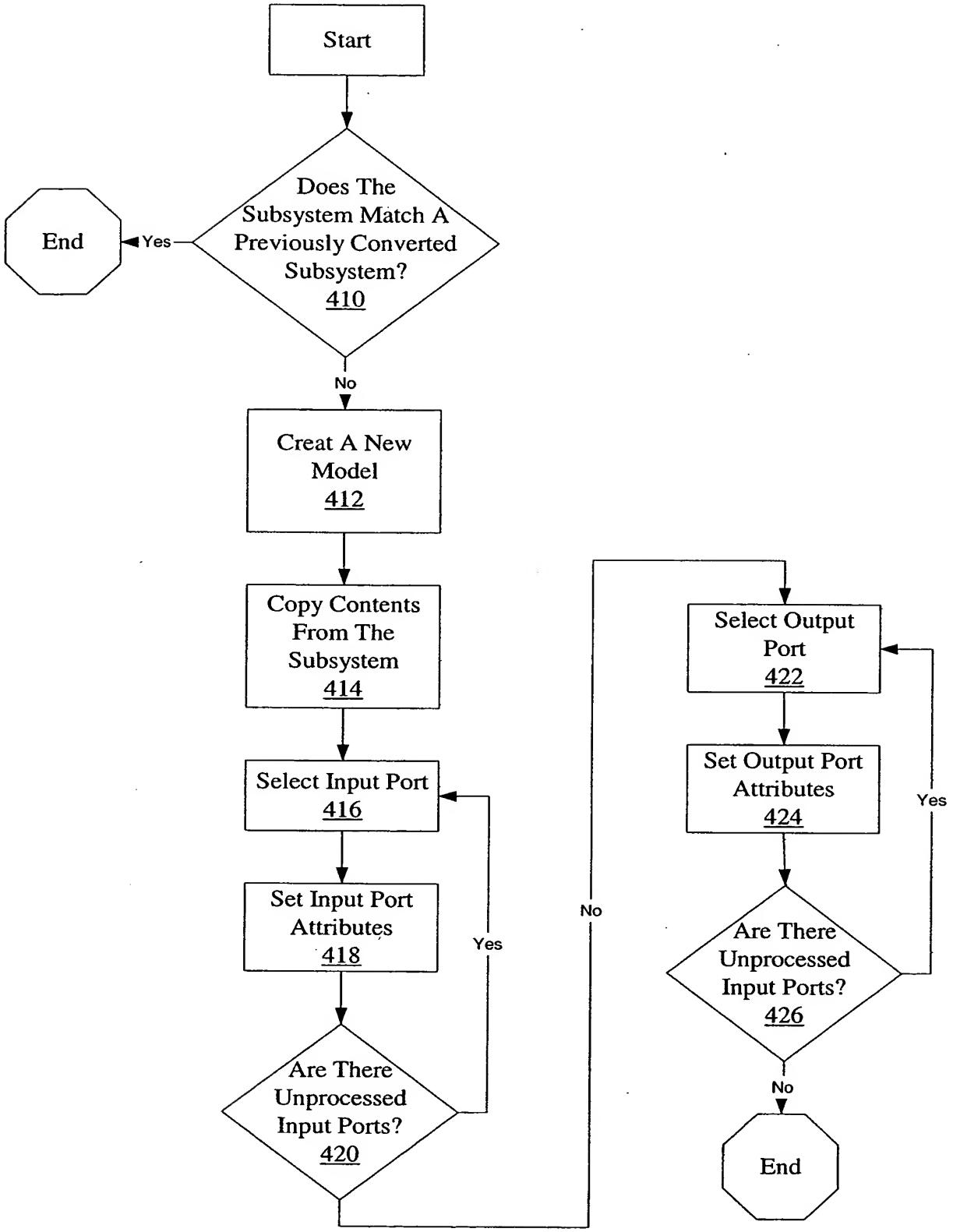
Start
Does The Subsystem Match A Previously Converted Subsystem? 410
Yes
End
No
Creat A New Model 412
Copy Contents From The Subsystem 414
Select Input Port 416
Set Input Port Attributes 418
Are There Unprocessed Input Ports? 420
Yes
No
Select Output Port 422
Set Output Port Attributes 424
Are There Unprocessed Input Ports? 426
Yes
No
End

FIG 22

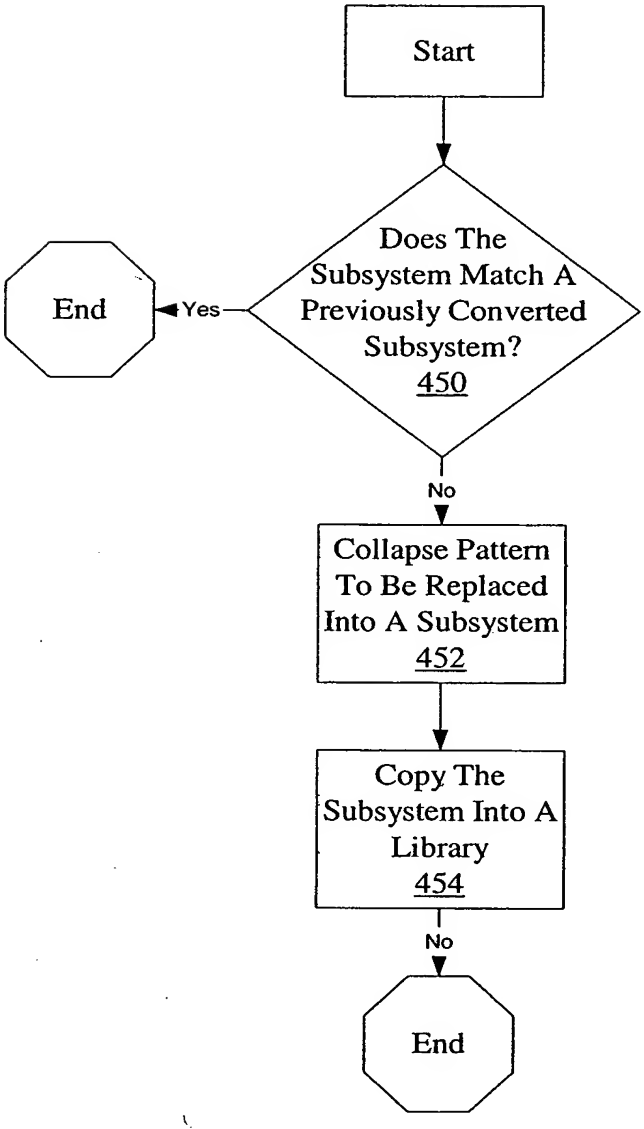
Start
Does The Subsystem Match A Previously Converted Subsystem?
450
Yes
End
No
Collapse Pattern To Be Replaced Into A Subsystem
452
Copy The Subsystem Into A Library
454
No
End

FIG 23

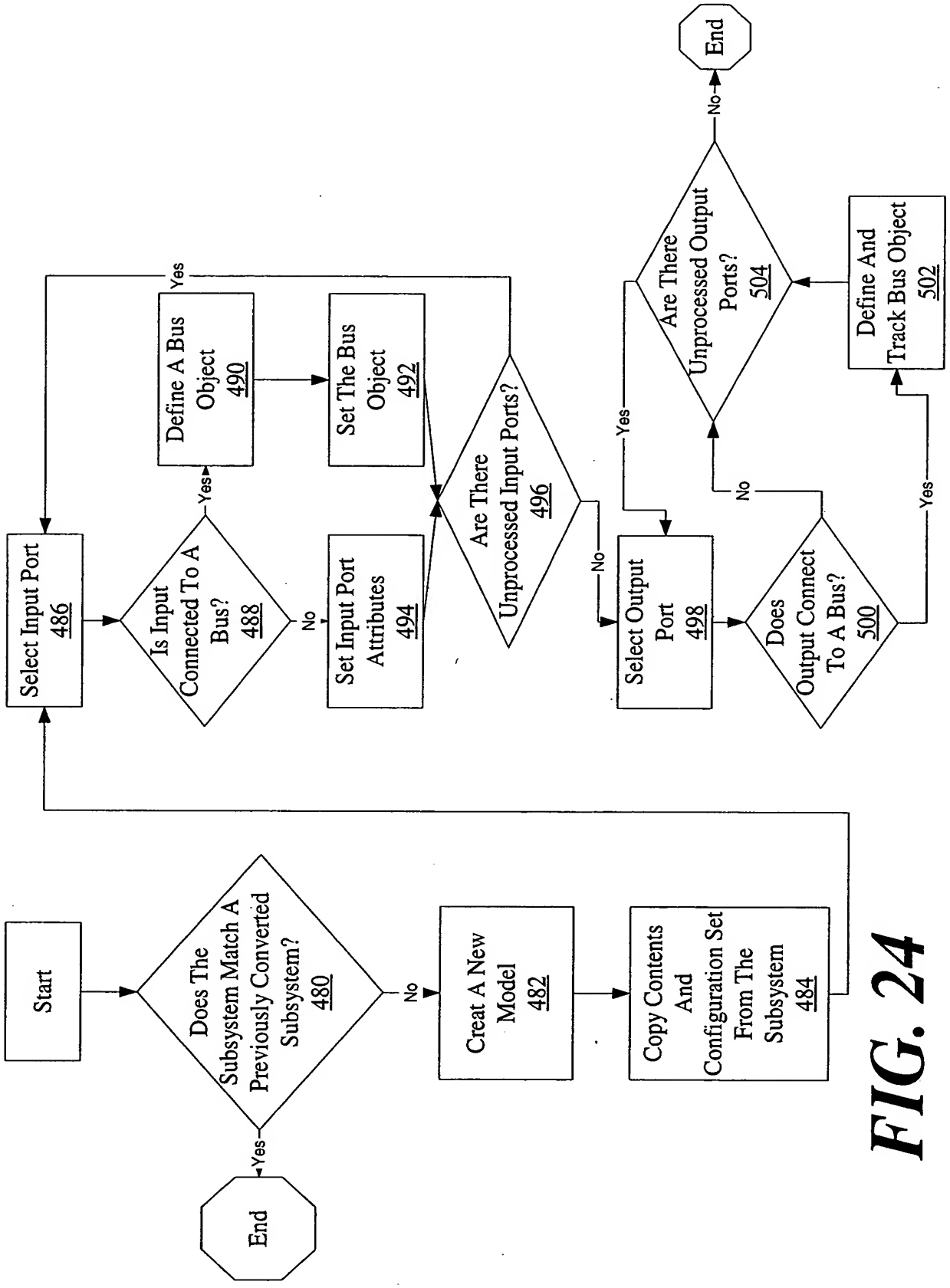
Start
Does The Subsystem Match A Previously Converted Subsystem? 480
Yes
End
No
Creat A New Model 482
Copy Contents And Configuration Set From The Subsystem 484
Select Input Port 486
Is Input Connected To A Bus? 488
Yes
Define A Bus Object 490
Set The Bus Object 492
No
Set Input Port Attributes 494
Are There Unprocessed Input Ports? 496
Yes
No
Select Output Port 498
Does Output Connect To A Bus? 500
Yes
Define And Track Bus Object 502
No
Are There Unprocessed Output Ports? 504
Yes
No
End

FIG. 24

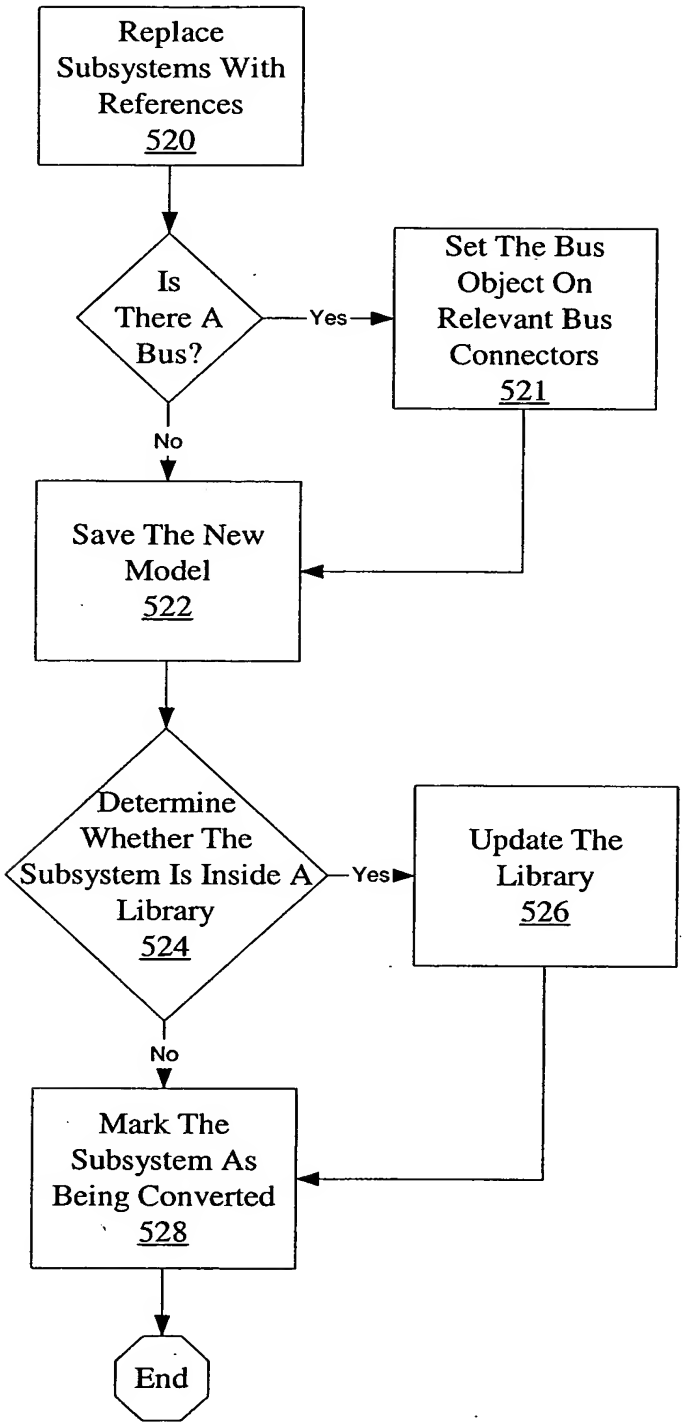
Replace Subsystems With References 520
Is There A Bus?
Yes
Set The Bus Object On Relevant Bus Connectors 521
No
Save The New Model 522
Determine Whether The Subsystem Is Inside A Library 524
Yes
Update The Library 526
No
Mark The Subsystem As Being Converted 528
End

FIG 25

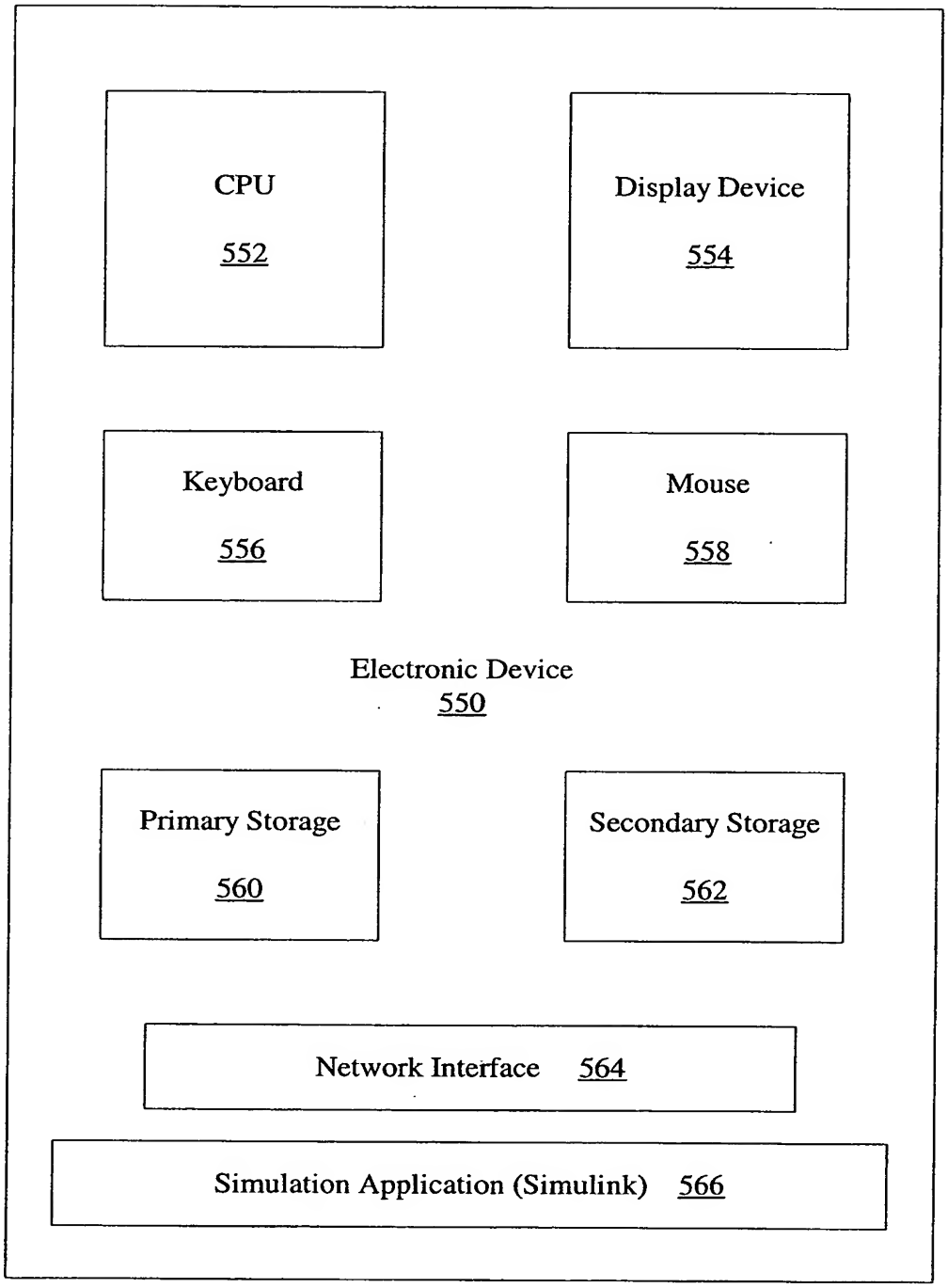
CPU
552
Display Device
554
Keyboard
556
Mouse
558
Electronic Device
550
Primary Storage
560
Secondary Storage
562
Network Interface 564
Simulation Application (Simulink) 566

FIG 26